

Design Configuration Subsystems Correctly and Distribute Safe Default Configurations

William L. Fithen, Software Engineering Institute [vita³]

Copyright © 2005 Carnegie Mellon University

2005-10-03

L4 / D/P⁴

Poorly designed configuration subsystems and poor default configurations may produce system vulnerabilities.

Description

The configuration of a system is the non-executable data delivered with a system that governs its dynamic behavior. The configuration is generally a set of variable values that supply information to the system in order to customize its behavior for a particular environment.

Default Configurations

This occurs when a system is shipped with a default configuration <define> <j2ee XML aspect oriented programming> that is insecure [Schneier 02: II. Default Configurations].

While this is not a programming vulnerability, it is an engineering vulnerability that is introduced in the product packaging phase of the software development life cycle.

Management or Debugging Interfaces Left Enabled

Administrative interfaces can lead to vulnerability. Sometimes the interface is left in by mistake. Sometimes it is intentional, but insecure.

In many cases, such administrative interfaces are configurable. In general, the solution to such interfaces is to configure them off. If, however, the product ships with such interfaces enabled by default, then one would reasonably classify this as a vulnerability in the product (if not in the software per se).

Administrative or management interfaces should always be restricted (via [authentication¹³, authorization¹⁴]) to proper administrators or managers.

Configuration Languages Too Complex

When the configuration "language" of a system is too complex, insufficiently expressive, contradictory, misleading, or ambiguous, it is reasonable to argue that this design will produce deployed systems that are vulnerable.

For example, avoid double or triple negatives, such as

```
no-read: false
```

When complex configuration languages are necessary,²⁰ be sure to include in system adequate tooling for creating, managing, and checking such configuration files.

3. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/320-BSI.html (Fithen, William L.)

13. <http://buildsecurityin.us-cert.gov/bsi/articles/knowledge/guidelines/321-BSI.html> (Use Authentication Mechanisms, Where Appropriate, Correctly)

14. <http://buildsecurityin.us-cert.gov/bsi/articles/knowledge/guidelines/322-BSI.html> (Use Authorization Mechanisms Correctly)

20. For example, J2EE deployment descriptors or Java Aspect Oriented Programming directives.

References

- [Landwehr 93] Landwehr, Carl; Bull, Alan; & McDermott, John. "A Taxonomy of Computer Program Security Flaws, with Examples." Technical report NRL/FR/5542--93/9591. United States Navy, Naval Research Laboratory, Nov. 1993.
- [Schneier 02] Schneier, Bruce. "Judging Microsoft." *Crypto-Gram Newsletter*. February 15, 2002. <http://www.schneier.com/crypto-gram-0202.html>
- [VU#247371] *Vulnerability Note VU#247371: Borland/Inprise Interbase SQL database server contains backdoor superuser account with known password*. cert.org, 2001. <http://www.kb.cert.org/vuls/id/247371>.
- [VU#602734] *Vulnerability Note VU#602734: Cisco default install of IBM Director agent fails to authenticate users for remote administration*. cert.org, 2004. <http://www.kb.cert.org/vuls/id/602734>.
- [VU#858726] *Vulnerability Note VU#858726: MailPost discloses sensitive system information when operating in debug mode*. cert.org, 2004. <http://www.kb.cert.org/vuls/id/858726>.

[1²⁶]

Carnegie Mellon Copyright

Copyright © Carnegie Mellon University 2005-2010.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

26. <file:///Users/wlf/Workspaces/Eclipse-3.1/swa-content/documents/html-upload/knowledge/guidelines/configuration.html#d0e119>

1. <mailto:permission@sei.cmu.edu>